Andrew Spate, Nicholas Spudich, Kevin Andor, Nathaniel DeHart

Vulcan’s Spire

Specifications Document

# Instructor Comments/Evaluation

Table of Contents

[Instructor Comments/Evaluation 1](#_Toc86933703)

[Abstract](#_Toc86933704) 4

[Purpose and Use 4](#_Toc86933705)

[Intended Audience 5](#_Toc86933706)

[System Description 5](#_Toc86933707)

[Overview 5](#_Toc86933708)

[Environment and Constraints 6](#_Toc86933709)

[End User Profile 6](#_Toc86933710)

[User Interaction 6](#_Toc86933711)

[Hardware Constraints 7](#_Toc86933712)

[Software Constraints 7](#_Toc86933713)

[Time Constraints 8](#_Toc86933714)

[Cost Constraints 8](#_Toc86933715)

[Other Concerns 8](#_Toc86933716)

[Acceptance Test Criteria 9](#_Toc86933717)

[Testers (who all will test) 9](#_Toc86933718)

[Criteria for User Acceptance 9](#_Toc86933719)

[Integration of Separate Parts and Installation 10](#_Toc86933720)

[System Modeling 11](#_Toc86933721)

[Functional: Use Cases & Scenarios 11](#_Toc86933722)

[Entity: Class Diagrams and names/descriptions/types 13](#_Toc86933723)

[Dynamic: Statechart 16](#_Toc86933726)

Statechart States Description [16](#_Toc86933728)

[Dataflow Scenario 19](#_Toc86933730)

[Components / Tools Needed 20](#_Toc86933731)

[Appendix: Glossary of Terms 20](#_Toc86933732)

[Appendix: Team Details 22](#_Toc86933733)

[Appendix: Writing Center report 22](#_Toc86933734)

[Appendix: Workflow Authentication 24](#_Toc86933735)

Abstract

Vulcan’s Spire is a non-traditional game that will provide users a fun introduction to the mythological figure of the Vulcan while inside the most iconic locations of the California University of PA campus. The driving focus of this game is to create an interactive way to attract potential students into the Computer Science Program, while developing a captivating narrative about the University’s beloved mascot, Blaze the Vulcan, as he completes challenges and wards off antagonists through his quest to find himself and save the beloved clocks at Old Main. The game will be targeted towards prospective and current students; however, it will be available to anyone who wants to play it. This document will outline the specifications to create Vulcan’s Spire, including aspects such as platform for end user, client testing, and a breakdown of the system.

## Purpose and Use

This Specification’s document is designed for the agreement of the developer and client, to align understandings and come to agreement on Vulcan’s Spire. Specifications within will include, but are not limited, to the game’s parameters and functionality. The client may dispute the contents of this document should it not follow their business model and/or their requirements. Developers and the clients will hereby discuss any and all modifications that must be made in accordance with delivery of the described product. Should the specifications be agreeable for both the client and development team, this document will serve as a binding contract for the duration of the project.

## Intended Audience

The intended audience of this Specification Document includes the clients and software developers. The clients will use this document to verify that Vulcan’s Spire meets all their desired attributes and functionality. If the client or development team wish to modify this contract after initial agreement, both parties must discuss the potential modifications, and the modifications must be ratified. The information contained herein will create the parameters of the application for the development team to follow. With constraints and restrictions given, the development team can begin the creation of Vulcan’s Spire that will run in accordance with this document. The diagrams and information within will help guide the development team’s efforts and keep perspective of the project along the path of creation.

# System Description

Overview

Vulcan’s Spire is a software application that seeks to provide users with an enjoyable and informative learning experience in the format of a game. The game will provide an intimate first-person experience with the main character, and California University of PA mascot, Blaze the Vulcan as he moves throughout a pseudo-realistic environment, engaging in combat and challenges to save his campus.

## Environment and Constraints

### End User Profile

Dependent on the medium of input-entry, users will need to learn the controls of the game. For example, if a controller is used the user must know what each key maps to\* within the game’s controls. The main forms of entry will be mouse & keyboard, controller, and touchscreen through phone apps. While some less-experienced users may experience a slight learning curve adjusting to the key commands or screen interface, it will feature controls that align with industry standards and norms for key placement and functionality.

### User Interaction (mouse, keyboard, monitor, printer)

Users will interact with the game through a Windows or Android device. Both the mobile and console applications will require an initial download and further patching may be necessary. Once this step is completed the user will launch it and can start playing the game either through a “New Game” or from a previous saved file, “Load Game”. The game controls can best be divided by which medium is used. For the console application, the game controls will primarily use the keyboard, with some menu interactions using the mouse for quicker access. This is the typical user experience for many games, and Vulcan’s Spire is no different. For the Android application, the user experience will be driven via the touch screen. Users will have an onscreen joystick/buttons to move the character, and buttons for the different actions.

### Hardware Constraints

The game will require a smartphone or a computer to play. This computer/smartphone will need to be using a current operating system that is supported with regular software updates. Legacy devices will not be supported. Specifics such as an upgraded graphics card are not an anticipated barrier of playing Vulcan’s Spire, which will allow for a broader audience to enjoy the game as opposed to others that need additional GPU consumption. Other constraints are ease of access, or preferability, such as using a controller or larger displays. In regard to the Android application, some things like a larger phone screen will create a more enjoyable experience, but Vulcan’s Spire will not require a specific model or screen size.

Base computer specifications:

|  |  |
| --- | --- |
| * 8 GB RAM * Storage size will be determined based on final executable released | * Intel I3 Processor or equivalent * Keyboard * Mouse * Display |

### Software Constraints (OS, compiler, other software running)

Computers must have a Windows operating system installed or use some form of virtual machine that can access it, such as BlueStacks (PC to Android), or VirtualBox (OS to Windows). Smartphones must be Android to be compatible with the software type (iOS will not be supported). Some older devices may experience slower framerates, but the application is designed to be usable on most devices with that in mind.

### Time Constraints

The time required to download the file depends on the specific user’s internet connection. Users will be subjected to the cap on download speed provided by their Internet service provider, or by wireless bandwidths through Wi-Fi downloading. Initial set-up of the installation may require additional time. As the file is a game, it will house a larger file which may require additional time.

### Cost Constraints

The development team would recommend the pricing of the Vulcan’s Spire to be a free project; however, this would be at the discretion of the client as plenty of games have pay to play components. Aside from the cost of the game, the end user will incur the soft-costs of owning a computer/smartphone and a data connection of internet/mobile data depending on the medium of access. While not a direct cost of Vulcan’s Spire, the hardware constraints could be costly given current market pricing (which we will not speculate because of market volatility regarding computers at the moment). Smart phone pricing for Android devices are slightly more financially reasonable, and again we will not speculate on costs at the time of completion.

### Other Concerns

The successful user will need to have a grasp of computing techniques to proceed through the game quickly. Users will need to spend time to learn the world and gather information to complete puzzles, defeat enemies, and reach the top of the clocktower.

## Acceptance Test Criteria

### Testers

Primary Testing will be the responsibility of the development team. After successful versions of Vulcan’s Spire are updated, the team intends to spend time reviewing the modifications from within the game setting, verifying all functionality and performance. While primary testing will be completed within the team, if time permits, a second wider range of persons not associated with development will be assembled for Beta testing Vulcan’s Spire. However, this testing would be after the main systems have been established and the game’s initial version has been successfully completed. All who test outside of the development team will be present or will follow up with the individual for feedback and/or criticism of the game experience.

### Criteria for User Acceptance

Our clients expect a satisfying and enjoyable playing experience. To declare this game enjoyable, we will be judging the game on a few criteria, such as reliability, satisfying physics/character interactions, enjoyable story, and smooth controls. These criteria will be achieved using the Unity Game designer interface. Unity combines endless possibilities with many defaults for items such as physics, with the ability to overwrite for a customizable and unique game experience. While many of the criteria for acceptance may seem ambiguous, we believe that thorough qualitative tests will provide ample feedback to fuel game modifications and updates.

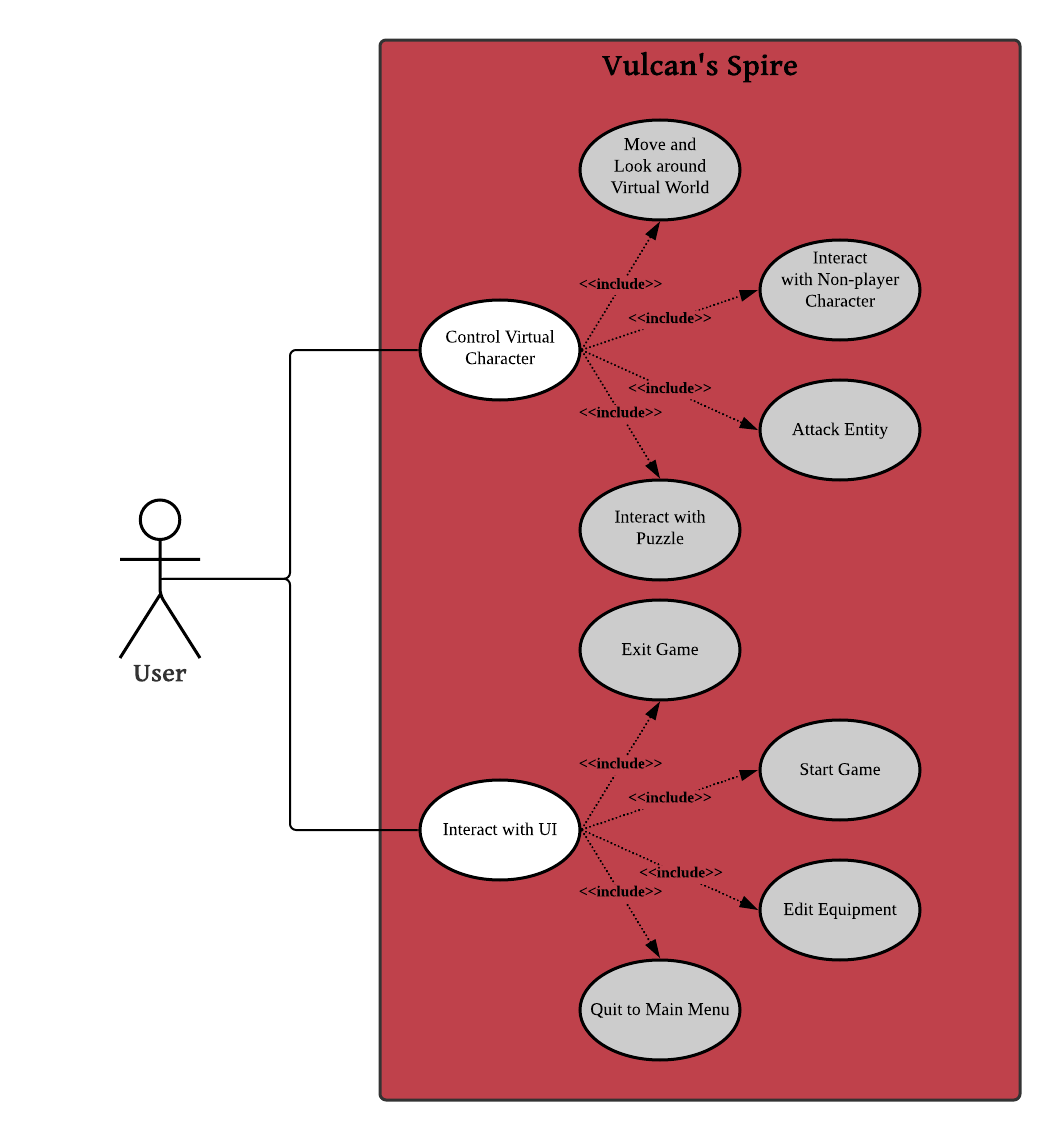
Aspects of the game in which we will be assessing include:

|  |  |
| --- | --- |
| * Vulcan’s Spire can be installed on any machine that meets the base system requirements * User can start a new game * User can load a previously saved game * Users can save their game * User can move Vulcan | * User can interact with game environment, including non-playable characters * Game will output acceptable experience * User interface will be intuitive and simplistic |

## Integration of Separate Parts and Installation

Vulcan’s Spire will be distributed as an executable file. Users will download the game and install will be straightforward for an average computer user. The game will be installed to the user’s local disk. To play the game, the user will need to run the executable, and a graphical user interface will then walk the user through anything else that needs completed within the game before play.

# System Modeling



*Figure 1*

*Control Virtual Character:*

* *Move and Look around Virtual World:* Walk around in a virtual 3D environment.
* *Interact with Non-Player-Character:* Prompt an NPC to display written dialogue.
* *Attack Enemy Entity:* Attack enemy characters using an equipped item.
* *Interact with Puzzle:* Complete puzzles to allow the user to move forward in progression.

*Interact with UI:*

* *Exit Game:* Closes the application.
* *Start Game:* Create a new save file or load previous data.
* *Edit Equipment:* Select UI icons to alter the player’s characteristics such as defense and attack power.
* *Quit to Main Menu:* Unloads the game world and loads the Main Menu.

Scenario:

1. The user interacts with UI to start the game

2. The user moves their player character towards the stairs in Old Main.

3. An enemy moves in front of the player.

4. The player attacks the enemy.

5. The enemy takes 3 damage.

6. The enemy attacks the player.

7. The player takes 10 damage.

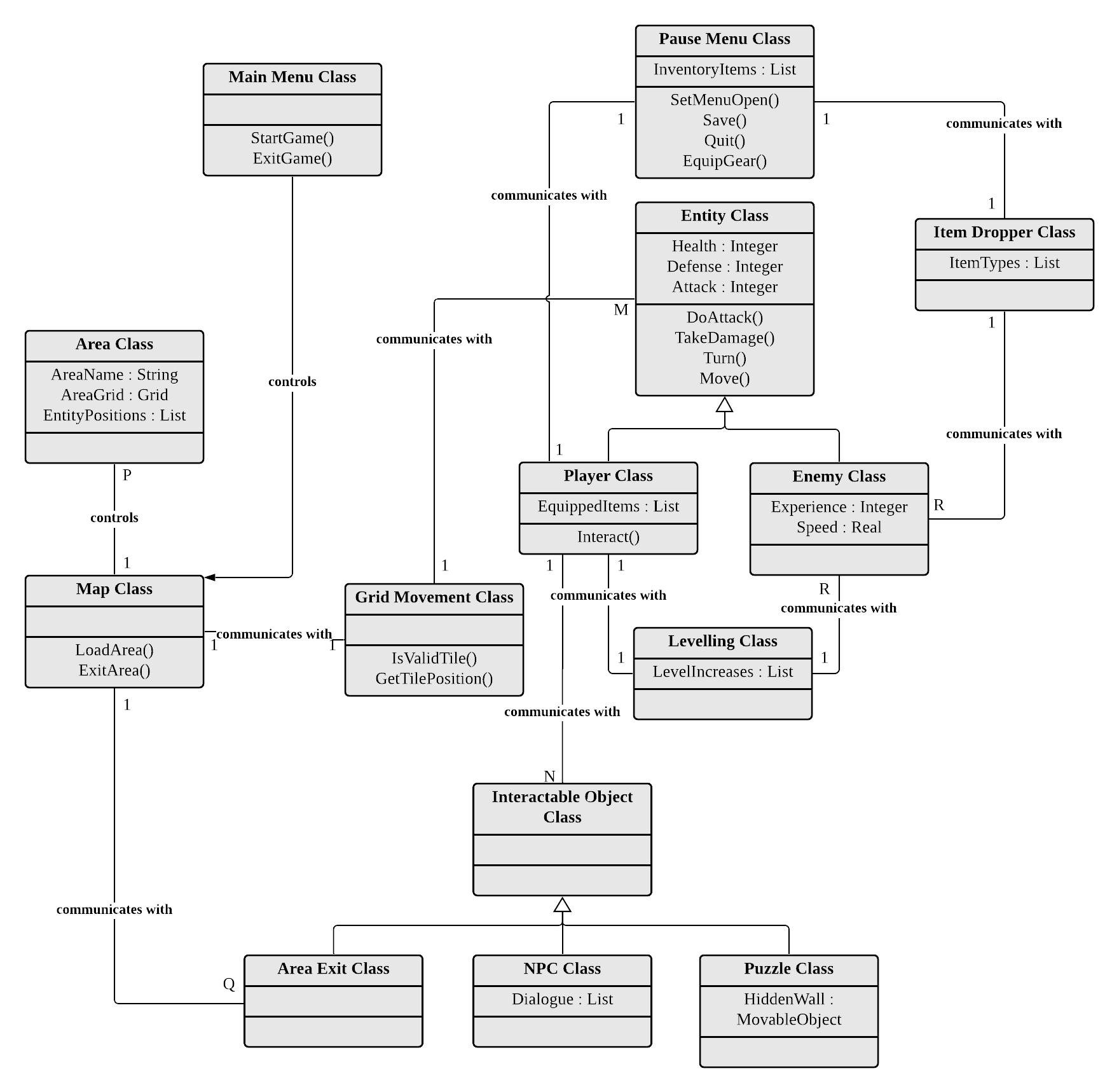
8. The player dies.

9. The last save game is loaded.

10. The player quits to the main menu.

11. The player exits the game.

## Entity: Class Diagram



*Figure 2*

### Class Diagram Description:

* *Main Menu Class:*

Gives the user the option to exit the game or start the game. The game can be started with the creation of new save data or by reloading old save data.

* *Pause Menu Class:*

Within the pause menu regular gameplay is paused, and the user may edit the statistics of the instance of *Player Class* by equipping different gear from within their inventory. There is also a button to quit to the main menu.

* *Entity Class:*

A base class with built in statistics like max health, attack power, and defense. Also has a connection to the *Grid Movement Class* for handling movement with the Move() function. Can attack other instances of the *Entity Class* using the DoAttack() function or take damage with the TakeDamage() function.

* *Player Class:*

Derives from the *Entity Class*. An instance of this class both moves and attacks based on the User’s input. The User may also perform inputs that call the Interact() function when an instance of *Interactable Object* is nearby the instance of this class.

* *Enemy Class:*

Derives from the *Entity Class*. An instance of this class both moves and attacks based on Artificial Intelligence.

* *Item Dropper Class:*

A class used to generate item drops for the player to add to their inventory. The item drops are based off a list of various possible items that may be instantiated.

* *Area Class:*

Contains the name of the area and a matrix for holding the area’s walkable grid. The grid is made up of tiles each of which holds their position and a Boolean value for telling whether the tile contains an obstacle or not.

* *Grid Movement Class:*

Receives calls from instances of the *Entity Class* that inquire about non-obstructed tiles around them and/or want to move to one of the tiles. Will initiate the movement for entities wanting to move to a nearby tile.

* *Map Class:*

Represents a few instances of *Area Class* on screen and contains functions to load or unload those respective areas.

* *Interactable Object Class:*

A base class for receiving “interact” calls.

* *Area Exit Class:*

Derives from *Interactable Object Class*. If interacted with, then this will tell the *Map Class* to unload the current area and load the virtual Cal U map.

* *NPC Class:*

Derives from *Interactable Object Class*. If interacted with, then this will print out written dialogue about this virtual world of Cal U and possibly hints on where to go next.

* *Puzzle Class:*

Derives from *Interactable Object Class*. If interacted with correctly, it will “actuate” a hidden wall allowing the player to progress further.

## Dynamic: Statechart

*Figure 3*

Statechart States Description

* *Main Menu:*

Gives the User the choice to either start the game or exit the game.

* *Gameplay Loop:*

Regularly updates the game logic at a fixed target number of times per second.

* *Check Player Inputs:*

Checks for User input changes via a keyboard, mouse, or touchscreen.

* *Pause Menu Loop:*

A loop separates from the gameplay loop for updating the inventory’s user-interface and allowing the player to choose to edit equipment.

* *Edit Equipment:*

Changes the player’s statistics (like defense and attack power) depending on the equipment being equipped/un-equipped.

* *Save Data:*

Saves the relevant data to a file for loading up when starting the game.

* *Attack Processing:*

Determines whether an attack hits an entity.

* *Damage Enemy:*

Causes an enemy to take damage with the amount of damage relating to the player’s attack power and the enemy’s defense.

* *Move Processing:*

Processes a request to move in a given direction from a given tile. If the entity that wants to move does not have an obstacle in front of it, then it will begin to move that entity to that tile.

* *Interaction:*

This is where the player will be able to interact with various objects in an open-ended fashion. A simple interaction with an area’s exit will transport the player outside, but puzzles and NPCs will have more in-depth interactions. Interactions will vary in the number of options that the user has according to what object is being interacted with.

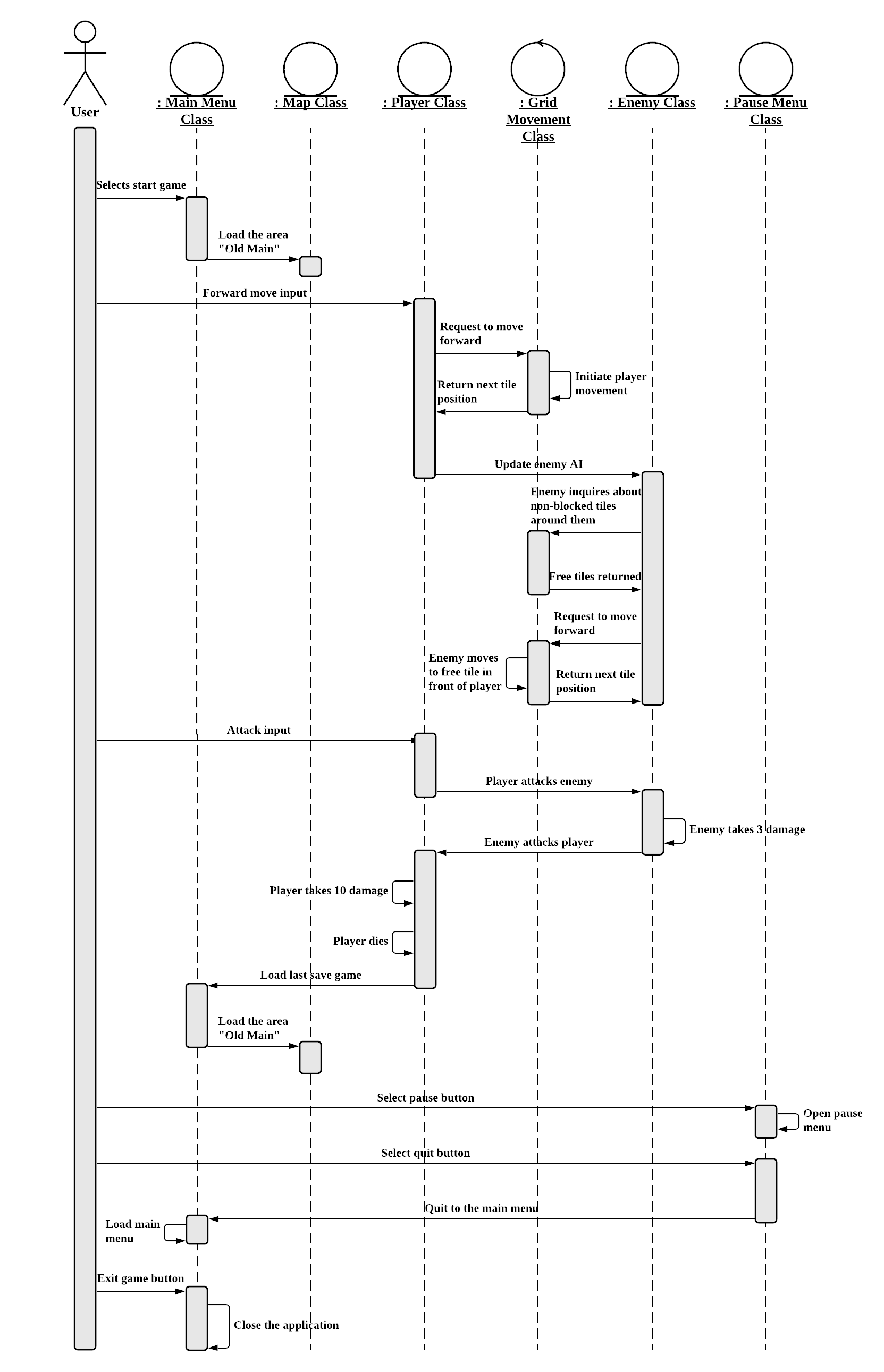
* *Update Enemy AI:*

Updates the AI of each active enemy. These AIs will essentially create the inputs for controlling the enemy entities’ movement and attacks.

* *Update Physics:*

Updates the movement of all entities and handles physical object collisions.

Dataflow Scenario



*Figure 4*

Components / Tools Needed

* A computer running Windows 10 will be needed to work with the game engine and run the game software, and an Android phone/tablet will be needed to test development for Android devices.
* Standard PC input devices such as keyboards and mice, as well as console game controllers will be needed as input for the game.
* The Unity game engine and C# libraries will be needed to write and compile the software.
* Blender will be used to create custom 3D assets for the game as needed.
* Graphic design software will be used to create custom UI assets for the game as needed

# Appendix: Glossary of Terms

**Virtual machine** - A computer that is emulated on another machine using software.

**BlueStacks** – An emulator that allows Android applications to be run on Windows and macOS.

**VirtualBox** – A cross-platform application that allows multiple instances of virtual machines running various operating systems within the same machine.

**Vulcan’s Spire** – The name and current title of the project.

**Unity** – A cross-platform engine primarily used to make both 2D and 3D games.

**Non-Player-Character (NPC)** – A character entity in a game that isn’t controllable.

**User Interface (UI)** – A graphical interface that the user can interact with.

**Save data** – Data that records the users progress in the game which can be loaded again at any time.

**Artificial intelligence** – A computer’s ability to act on it’s own and perform human tasks.

**Player characteristics/statistics** – The player character’s attributes used for various numerical calculations in the game.

**Item** – An object in the game the player can collect and store in their inventory for later use.

**Player inventory** – A collection of the user’s obtained items in the game which the player can access and use.

**Equipment** – Items that can be used to affect player stats in the game.

**Enemy** – An entity in the game made to oppose and attack the player when both entities come in contact with each-other.

**Puzzle** – Interactive knowledge challenges made to oppose the player.

**Tiles** – The main unit of movement that all mobile entities abide by.

Appendix: Team Details

The workflow leader is Nathaniel DeHart. The document was divided into sections for each team member to complete. Any member who had questions or concerns voiced them using Discord. Each member checked-in upon completion to give summary of their contributions. This ensured a deeper and clearer understanding of the work to be performed. Upon completion of this document, each member read through and gave their “okay” for the document to be pushed forward to the writing center and submitted.

# Appendix: Writing Center report

Graphical user interface, text, application, email

Description automatically generated

# Appendix: Workflow Authentication

I, Nicholas Spudich, hereby confirm that I have contributed to the information outlined in this document.

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I, Andrew Spate, hereby confirm that I have contributed to the information outlined in this document.

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I, Kevin Andor, hereby confirm that I have contributed to the information outlined in this document.

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I, Nathaniel DeHart, hereby confirm that I have contributed to the information outlined in this document.

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_